A track finding algorithm for the EicC central detector*

Zulun Zhang,[1] Guoming Liu,[1] Yutie Liang,[2,3] and Aiqiang Guo[2,3,†]

[1]*Key Laboratory of Atomic and Subatomic Structure and Quantum Control (MOE), Guangdong Basic Research Center of Excellence for Structure and Fundamental Interactions of Matter, Institute of Quantum Matter, South China Normal University, Guangzhou 510006, China.*
[2]*Institute of Modern Physics, Chinese Academy of Sciences, Lanzhou 730000, China*
[3]*University of Chinese Academy of Sciences, Beijing 100049, China*

This paper presents a track finding algorithm designed for track reconstruction in Electron-ion collider in China (EicC). The goal of algorithm is to fulfill the criterion of high track reconstruction efficiency. The algorithm is modularly constructed, leveraging an advanced cellular automaton model and the Kalman filter method to implement its core functionality. We optimized the algorithm based on the Monte Carlo events with full detector simulation based on the EicCRoot software framework. The performance of the method is also validated and excellent track reconstruction efficiency is obtained, which can well meet the physical requirements of the EicC experiment.

Keywords: Track finding, Electron-ion collider in China, Cellular automaton, Kalman filter

I. INTRODUCTION

Lepton scattering is an established ideal tool for studying the inner structure of nucleons. As a future high-energy nuclear physics project, EicC has been proposed [1]. The primary objectives of the EicC will include conducting precision measurements of the nucleon's structure in the sea quark region, performing 3D tomography of nucleons, exploring the partonic structure of nuclei, and investigating how partons interact with the nuclear environment. Additionally, the EicC will also focus on studying exotic states [2], particularly those containing heavy flavor quarks.

The center of mass energy of the EicC will range from 15 GeV to 20 GeV, with the luminosity higher than $2.0 \times 10^{33}\, cm^{-2} \cdot s^{-1}$. Driven by the physics program of EicC, a conceptual design for a general-purpose spectrometer is proposed [3], which has a cylindrical structure, built with different layers around the beam pipe. The charged particles produced first enter a tracking detector, leaving traces in the sensitive electronics. Hits are used to reconstruct charged particles' trajectories and their origins. The layout of the vertex and tracking system is depicted in Fig.1. The central tracking system is divided into three regions: barrel region, ion-going region, and electron-going region. The tracking detector in each region is described as follows.

The tracking detector in the barrel region consists of an inner silicon layer and an outer micropattern gaseous detectors (MPGD) layer. The inner silicon cylinder has three vertex layers and two tracking layers, occupying an area with a maximum radius of 15 cm and a total length of 28 cm. The vertex layer utilizes wafer-scale suture sensors that bend around a beam pipe made of a beryllium cylinder with a radius of 3.17 cm, and the tracking layers also use the same stitched sensors but with different support structure. The outer MPGD has two closely-spaced 2-D layers of Micromegas which are chosen to cover the outermost barrel region. Their mean radii are approximately 65 cm and 67 cm, and their maximum total length of approximately 200 cm.

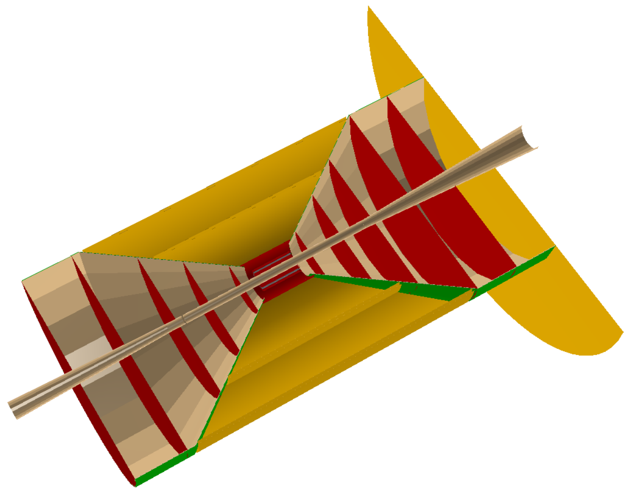

Fig. 1. The conceptual design of EicC tracking system.

In the ion-going direction, the five silicon disks start at the point of interaction 25 cm along the z-axis and extend to 134 cm. The minimum radius is determined by the divergence of the beam and tube, while the maximum external radius is about 67 cm. The Micromegas detector is located 165 cm in front, with an inner radius of about 8 cm and an outer radius of about 150 cm.

The electron-going direction has five silicon disks. These disks start at 25 cm along the z-axis from the point of interaction and extend back to 145 cm. The minimum radius of the disc is determined by the divergence of the beam tubes, which ensures that they do not interfere with the beam path. The maximum outer radius of the disc is about 67 cm, providing ample coverage to track particles in the receding area.

* This work is supported by National Natural Science Foundation of China (NSFC) under Contracts Nos. 12375194; Guangdong Major Project of Basic and Applied Basic Research No. 2020B0301030008.

† Corresponding author: Email: guoaq@impcas.ac.cn, Address: 509 Nanchang Rd., Lanzhou 730000, China, Contact No.: 18693271981.

II. TRACK RECONSTRUCTION FOR EICC

Track reconstruction refers to the process of identifying and reconstructing the trajectories of charged particles as they pass through a particle detector. Charged particles, e.g. electrons, interact with the material in a detector and leave behind signals (e.g. ionization or light). These signals are recorded at specific hits in the detector, typically using layers of sensors arranged in a geometric pattern. Track finding involves analyzing the spatial distribution of these signals (hits) to determine the path of the particle. The trajectory of the particle is usually a curved path because of the presence of a magnetic field, which exerts a force on the moving charged particle. We can get the key properties of the particles, such as momentum, charge, vertex, etc., by fitting the hits belonging to one track utilizing the Kalman Filter(KF) method [4].

A. The process of track reconstruction for EicC

Leveraging the geometric structure of the EicC, we have developed a tracking algorithm that integrates cellular automaton (CA) [5] with the KF method. This algorithm processes tracks generated by Monte Carlo (MC) simulations. Utilizing the EicCRoot software framework, the algorithm reconstructs tracks based on the hits left by simulated tracks on the detector layers. The implementation process of the algorithm is shown as follows:

- Read all the hits information from the simulation.
- Perform the track-finding algorithm by the CA method.
- Fit the found track candidates by the KF method to obtain the track information.
- Validate the performance of the algorithm by comparing the reconstructed and MC truth information.

CA are mathematical models composed of a grid of cells, each of which can exist in a finite number of states. The state of each cell is updated based on a set of rules that consider its current state and the states of its neighboring cells. Compared to traditional track finding methods, e.g. Hough transformation, one of the most significant advantages of CA is their inherent parallelism. In track finding, where large amounts of data from detectors need to be processed simultaneously, the ability to process many elements in parallel significantly reduces the overall computation time, and CA algorithms can be efficiently implemented on parallel hardware, such as Graphics Processing Units or dedicated parallel computing clusters.

The KF is an optimal estimation algorithm used to predict and correct the state of a dynamic system over time, based on noisy or incomplete measurements. It operates recursively by combining prior knowledge (predictions) with new data (observations) to improve accuracy in estimating unknown variables [6]. In tracking particles in detectors, its ability to handle noisy measurements, estimate the state of a system over time, and optimize trajectory reconstruction makes it highly effective in various experimental scenarios.

B. The application of CA to EicC track finding

As mentioned before, the tracking system design of EicC is divided into three parts: barrel region, ion-going region, and electron-going region. As shown in Fig. 2, the barrel part has nine layers, the ion-going end cap (the positive z direction) has six layers, and the e-going end cap has five layers. The layer IDs in barrel are defined as: 0.1.2.3.4.5.6.7.8, in the ion-going end cap: 9.10.11.12.13.14 and in e-going end cap: 15.16.17.18.19. The whole track-finding procedure is illustrated below.

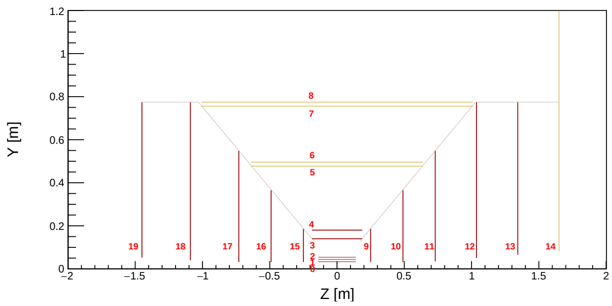

Fig. 2. The profile of EicC tracking system and the definition of the layer ID.

Table 1. The statistical distribution of the graph list

Graph list	Frequency
0.1.2.3.4.5.6.7.8	5432
0.1.2.9.10.11.12.13.14	2168
0.1.2.15.16.17.18.19	1520
10.11.12.13.14	480
15.16.17.18.19	400

- Defination of graph
  A graph is a data structure that encodes information about all detector layers, the pairs of adjacent layers, and the root layer through which the particle travels. The construction of the graph is a crucial step in the algorithm's initialization process, with all subsequent algorithmic operations building the necessary data structures based on the specific layer list of each graph.

  Using the EicCRoot software framework [7], we generated 100,000 MC single-muon events, where the track momentum ranges from 1 to 5 GeV and the angular distribution covers the full 0 to 360 degrees. We meticulously analyzed all track trajectories and recorded every possible combination of detector layers that a track can pass through, as shown in Table 1. These combinations, referred to as graphs, form the basis for the subsequent track-finding algorithm.

- Creation of cells

After graph creation, we need to connect the hit points of adjacent layers to form a doublet, which serves as a cell in the graph. The core component of the algorithm involves determining whether two hits from adjacent layers can be linked to form a doublet. Given that a particle's trajectory in the tracking system is helical, we decompose the trajectory into two planes: the x-y plane and the r-z plane. On these two planes, we evaluate whether the angle formed by connecting the two hit points to the coordinate origin meets a predefined critical value, thereby determining if hits between adjacent layers can be linked to form a doublet. We analyze 10,000 events generated by MC simulation and examine the angles formed by adjacent hit points on each real track. According to this study, on the x-y plane (θ_{x-y}), 99.53% of the included angles were less than or equal to 2.618 mrad, and 100% were less than or equal to 4.363 mrad. On the r-z plane (θ_{r-z}), 97.90% of the angles were less than or equal to 0.0873 mrad, while 99.85% were less than or equal to 0.026 mrad. The result is shown in Fig. 3, 4 .

All adjacent layers in a graph are given to a function in turn, and the algorithm judges all hit pairs according to the geometrical requirement obtained from the simulation data shown above. Then all the hit pairs that satisfy the requirement are saved to a specific data structure for further processing.

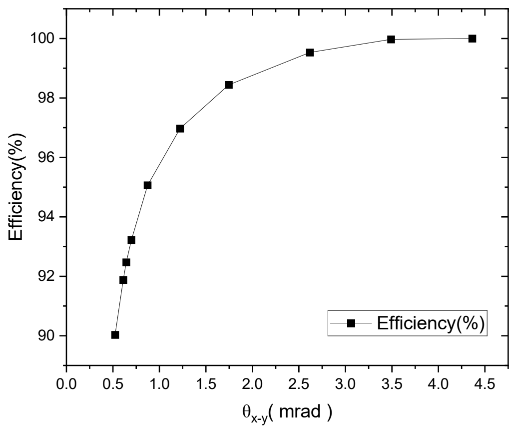

Fig. 3. The efficiency as a function of requirement on θ_{x-y}

- Cells connection

The connection of cells is the key procedure for track finding with CA. The first step is to convert all doublet data structures into cell data structures. Starting from the root layer of each graph, a state variable, CAState, is assigned to the each cell, initialized to zero. The second step consists of finding neighbors for each Cell. Two cells are considered neighboring cells if all of the following conditions are satisfied: Firstly, they belong to different layer pairs. Secondly, they share a common hit, where the inner hit of one cell is the outer hit of the other. Finally, the corresponding constraints are satisfied in both the x-y and r-z planes. The angle between two neighboring cells in $x-y$ plane is illustrated in Fig. 5.

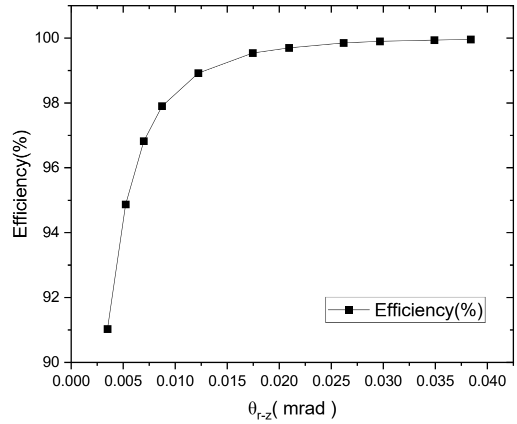

Fig. 4. The efficiency as a function of requirement on θ_{r-z}

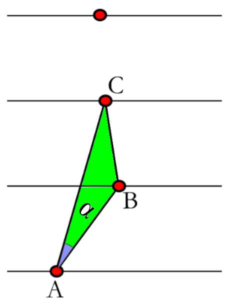

Fig. 5. The α_{x-y} of two neighboring cells in $x-y$ plane.

To establish the criteria for selecting neighboring cells in the x-y and r-z planes, we simulated 10,000 events. From the analysis of these MC events, we determined that in the r-z plane (α_{r-z}), 99.79% of the neighboring cells originating from a single track had an angle less than or equal to 0.000873 mrad, and 100% had an angle less than or equal to 0.00174 mrad. In the x-y plane (α_{x-y}), 99.92% of the neighboring cells had an angle less than or equal to 0.607 mrad and 99.96% had an angle less than or equal to 1.309 mrad, and all the result is shown in Fig. 6, 7. For each graph, the algorithm evaluates adjacent cells based on these criteria derived from the simulation. The IDs of matched cells are stored to ensure that matched adjacent cells can be accurately

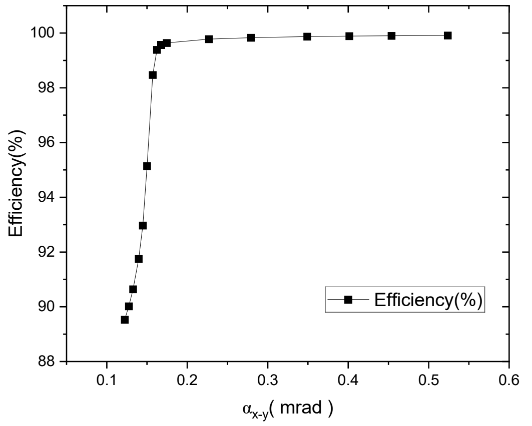

Fig. 6. The efficiency as a function of requirement on α_{x-y}

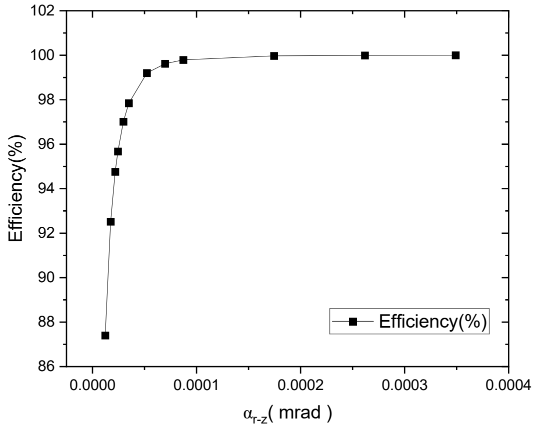

Fig. 7. The efficiency as a function of requirement on α_{r-z}

identified in subsequent stages of the algorithm.

- Evolution and track candidate creation

After establishing the graph, including all cells and their relationships, the final step in track finding is to select the longest path from a root cell within the graph. This is accomplished by evolving the graph over several generations according to a specific rule, allowing the longest path to be identified based on the state values of the cells. Initially, the state of each cell in the graph is set to zero. During the evolution process, the state values of all cells are updated based on the state value of the cell under investigation and its neighbors. A cell's state value is incremented by one if it matches the state value of any of its neighbors. The algorithm begins at the root layer of each graph and iterates over all the layer pairs within it. The total number of cycles is determined by the number of layers minus two in the graph. Figure 8 illustrates the state values of all cells for a four-layer graph after two cycles of evolution.

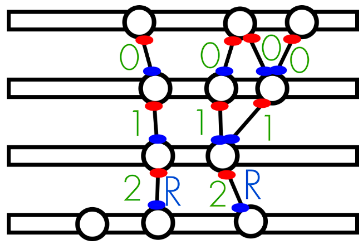

Fig. 8. The state values of all cells for a four-layer graph after two cycles of evolution.

After the evolution process, the algorithm conducts a depth-first search starting from a root cell to generate track candidates. A cell is identified as the root cell of the graph if its state value equals the total number of layers in the graph minus two. The entire graph is then traversed, and sequential cells with descending state values are selected as track candidates. These candidates are subsequently stored for further analysis and the track finding procedure is completed.

C. Kalman filter in EicC track reconstruction

With the track candidates, which represents subsequently stored hits information detected by the tracking detector, the track information can be extracted by fitting the track candidates with KF method. The KF represents an iterative and adaptive process designed to estimate the states of dynamic systems. It can be employed in track reconstruction on the assumption that the track can be regarded as a discrete dynamic system. In the fitting, the state of the track at each detector surface i is characterized by the state vector $\vec{p}_i$. Given the state vector $\vec{p}_{k-1}$, which delineates the state of the track at surface $k-1$, the system equation defines the propagated state vector $\vec{p}_k$ at the subsequent surface k. The KF utilizes both previous and current measurements to ascertain the current state.

In implementing the KF for EicC's track fitting algorithm, the process begins by constructing a measurement plane for each hit in a track candidate. Assuming the interaction point is at the coordinate origin, the momentum and direction of the first hit closest to the origin are used as initial conditions for the track representation. The track representation is then updated at each measurement plane using the KF method.

The track fitting involves iterating in two opposite directions, a process known as the smoothing procedure, to obtain the best estimation of track parameters. This bidirectional fitting helps refine the track estimate by incorporating information from both forward and backward passes, thus improving accuracy.

In the track fitting approach for EicC, it's common for the same root cell to have multiple candidate tracks, particularly

when hit points from different tracks are in close proximity. To address this, the algorithm selects the best track candidate based on the smallest chi-square value obtained during the KF fitting procedure. The chi-square value serves as a measure of fit quality, allowing the algorithm to identify the track candidate that best represents the observed hits with the least statistical deviation, ensuring the most accurate track reconstruction.

D. Algorithm optimization for track finding

The criteria used in the cell creation and connection are optimized according to simulated events. We simulated 10,000 events and analyzed these events by our algorithm. The tracks in simulation have a momentum of [0-1] GeV and an angle between [20-160] degree. The track multiplicity is five in the simulation. The criteria in the optimization include: θ_{x-y} and θ_{r-z} are the angles between two vectors formed by the hit and the collision point in different planes in constructing a doublet; α_{x-y} and α_{r-z} are the angles between two cells in different planes when connecting two cells as illustrated Fig. 5. The optimized results are shown below:

- Figure 9 shows the efficiency as a function of requirement on θ_{x-y}. From this study, the optimal value of parameter θ_{x-y} is 0.8727 mrad.

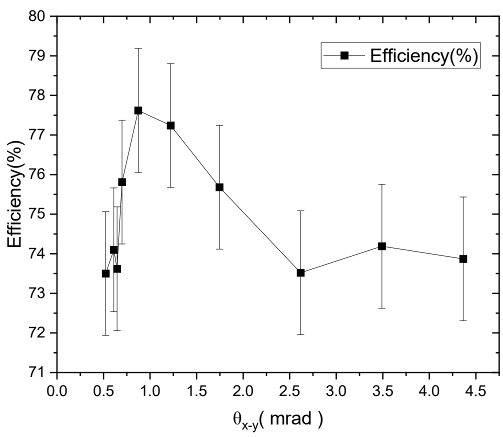

Fig. 9. Tracking efficiency as a function of requirement on θ_{x-y} parameter for the cell creation.

- Figure 10 shows the efficiency as a function of requirement on θ_{x-y}. From this study, the optimal value of parameter θ_{x-y} is 0.0122 mrad.

- Figure 11 shows the efficiency as a function of requirement on α_{x-y}. From this study, the optimal value of parameter α_{x-y} is 0.4537 mrad.

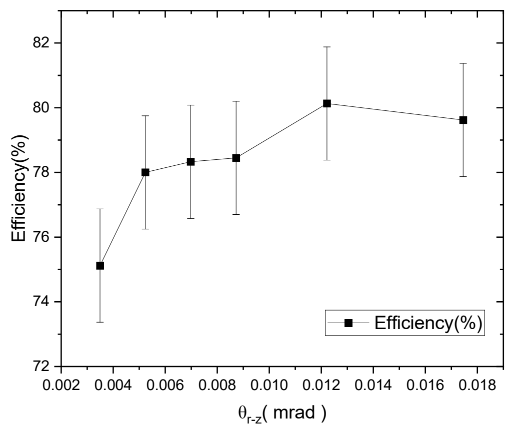

Fig. 10. Tracking efficiency as a function of requirement on θ_{r-z} parameter for the cell creation.

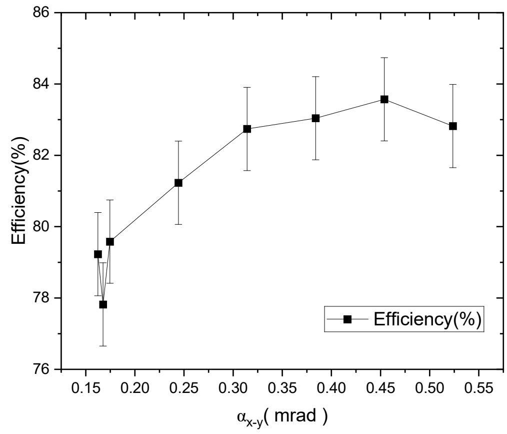

Fig. 11. Tracking efficiency as a function of requirement on α_{x-y} parameter

- Figure 12 shows the efficiency as a function of requirement on α_{r-z}. From this study, the optimal value of parameter α_{r-z} is 0.00014 mrad.

As depicted in the figures, due to the small curvature radius of the motion trajectory of low-momentum particles, the momentum chosen for our optimization process falls within a very low range. Consequently, the track efficiency resulting from this optimization is relatively low.

III. ALGORITHM PERFORMANCE

The main criteria for the quality of the track reconstruction that can be used to quantify the performance in tracking procedures are:

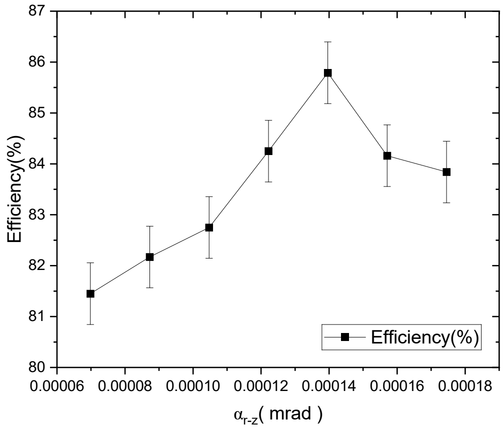

Fig. 12. Tracking efficiency as a function of requirement on α_{r-z} parameter

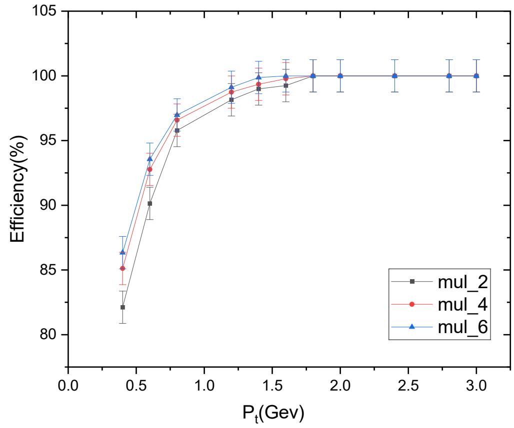

Fig. 14. The tracking efficiency of all reconstructed tracks.

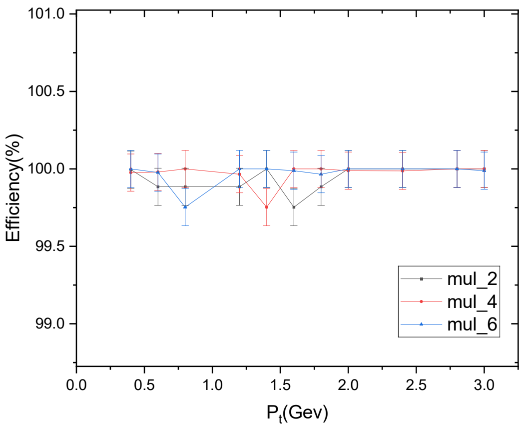

Fig. 13. The hit efficiency of all reconstructed tracks.

- **Hit efficiency**
  The hit efficiency, ϵ^{hit}, indicates the fraction of the reconstructed hits, N_{rec}^{hit} divided by the number of generated hit in a track, N_{gen}^{hit}.

$$\epsilon^{hit} = \frac{N_{rec}^{hit}}{N_{gen}^{hit}}$$

- **Tracking efficiency**
  The tracking efficiency, ϵ^{track}, indicates the fraction of the reconstructed tracks, N_{rec} divided by the number of generated track, N_{gen}.

$$\epsilon^{track} = \frac{N_{rec}^{track}}{N_{gen}^{track}}$$

- **Execution time**
  The execution time is the CPU time spent in reconstructing tracks.

The hit and tracking efficiencies are evaluated across different momentum ranges and track multiplicities using simulated events. The results of these studies are presented in Figs 13, 14, and 15. As we can see, the hit efficiency exhibits a near-perfect performance, achieving almost 100% across diverse momentum ranges. Furthermore, the trajectory reconstruction efficiency demonstrates a gradual ascent in tandem with an increase in momentum. Notably, there is a substantial surge in momentum from 0.4 GeV to 1.4 GeV, and when momentum surpasses 1.5 GeV, the efficiency nears perfection, approaching 100%. To successfully reconstruct a track, we require that at least three hits be found. Consequently, the efficiency at low P_t is worse because of the minimum P_t threshold needed for a track to reach the outer layer of the detector.

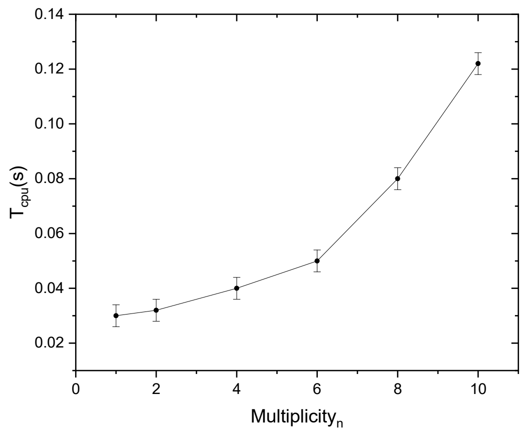

Fig. 15. The execution time of cpu with different multiplicity.

The CPU execution time escalates accordingly as the increasing of number of tracks multiplies. This augmentation in track multiplicity also leads to an increase in multi-threading overhead, resulting in a progressively more pronounced growth in execution time.

IV. SUMMARY

This paper presents a track reconstruction algorithm for the EicC central detector, developed using a combination of CA and KF methods. The track-finding process is initiated with the CA method, where hits in neighboring layers are connected based on geometrical information using graph theory. The longest path is then selected by evaluating the state values of each cell after the graph's evolution. Once track candidates are identified, they are fitted using the KF method to extract detailed track information, such as momentum, charge, and vertex position. The performance of this algorithm has been validated through simulations, demonstrating its efficiency and accuracy in meeting the physics requirements of the EicC project.

[1] D. P. Anderle *et al.*, Front. Phys. (Beijing) **16**, 64701 (2021), 2102.09222.
[2] X. Cao, Front. Phys. (Beijing) **18**, 44600 (2023).
[3] D. P. Anderle, A. Guo, F. Hekhorn, Y. Liang, Y. Ma, L. Xia, H. Xing, and Y. Zhao, Phys. Rev. D **109**, 034021 (2024).
[4] R. Fruhwirth, Nucl. Instrum. Meth. A **262**, 444 (1987).
[5] S. Wolfram, Rev. Mod. Phys. **55**, 601 (1983).
[6] R. Mankel, Rept. Prog. Phys. **67**, 553 (2004).
[7] M. Al-Turany, D. Bertini, R. Karabowicz, D. Kresan, P. Malzacher, T. Stockmanns, and F. Uhlig, Journal of Physics: Conference Series **396**, 022001 (2012).